UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
10/665,305	09/22/2003	Thomas Goering	11884-400201	5450

53000   7590   10/02/2009
KENYON & KENYON LLP
1500 K STREET N.W.
WASHINGTON, DC 20005

EXAMINER
DESAI, RACHNA SINGH

ART UNIT	PAPER NUMBER
2176	

MAIL DATE	DELIVERY MODE
10/02/2009	PAPER

Please find below and/or attached an Office communication concerning this application or proceeding.

The time period for reply, if any, is set in the attached communication.

RECORD OF ORAL HEARING

UNITED STATES PATENT AND TRADEMARK OFFICE

3   _____

4   BEFORE THE BOARD OF PATENT APPEALS

5   AND INTERFERENCES

6   _____

7   *EX PARTE* THOMAS GOERING

8   _____

9   Appeal 2009-002814
10   Application 10/665,305
11   Technology Center 2100
12   _____

13   Oral Hearing Held: September 9, 2009

14   _____

15   Before JOHN A. JEFFERY, ST. JOHN COURTENAY, III, and STEPHEN
16   C. SIU, *Administrative Patent Judges.*
17

18

19   APPEARANCES:

20   ON BEHALF OF THE APPELLANT:

21   JASON D. HARRIER, ESQUIRE
22   Kenyon & Kenyon, LLP
23   1500 K Street, N.W.
24   Washington, D.C. 20005
25
26
27
28
29
30
31
32

1   The above-entitled matter came on for oral hearing on Wednesday,

2   September 9, 2009, at The U.S. Patent and Trademark Office, 600 Dulany

3   Street, Alexandria, Virginia, before Dawn A. Brown, Notary Public.

4

5   THE USHER: Calendar Number 13, Appeal Number 2009-2814.

6   Mr. Harrier.

7   JUDGE JEFFERY: Okay. Thank you. Good morning.

8   MR. HARRIER: Good morning.

9   JUDGE JEFFERY: Welcome. You have 20 minutes, and you can

10 begin when you're ready.

11   MR. HARRIER: All right. Thanks a lot.

12   May it please the Court, my name is Jason Harrier with the law firm of

13 Kenyon and Kenyon, and we represent the Appellant, Thomas Goering, and

14 SAP AG in this appeal.

15   I assume that you are all familiar with our application and cited

16 reference. So if I may, I'll just go ahead with our arguments.

17   The first thing that I would like to discuss is an important distinction

18 between our claimed subject matter and the form data that is present in the

19 cited reference, Hitchcock.

20   Our claimed subject matter involves an output module. In the

21 Examiner's Final Rejection and within their Brief, they've argued the output

22 module is analogous to the data that is entered into a form and shared within

23 the system that is disclosed by Hitchcock.

24   It is our contention that our output module -- it is actually an executable

25 module that is used to generate and present the form. So whether it would be a

1   presentation to a printer, so that it can be printed, or a presentation to a web

2   page, that it can be viewed, our output module is responsible for taking form

3   data, which includes these reusable form elements that we can use within a

4   plurality of different forms. We take that and it actually is responsible for

5   presenting the form.

6   The distinction that I'd like to draw between that and the data in

7   Hitchcock is the data in Hitchcock, the example that they use in the reference

8   would be a college application process where a student would put in data into a

9   college application and that would be uploaded to maybe a third-party server.

10   When the student subsequently applies to another university, an

11   application with similar data, the data may be pulled down from the first

12   application and sucked down into the second application so it doesn't need to

13   be retyped.

14   But that form data is different than the actual form. The data that is

15   being shared between different forms in the reference is actually the data that

16   is entered into the form after it has been presented to a user, whereas our

17   output module is responsible for the actual presentation of the form.

18   JUDGE JEFFERY: Well, I think that is the crux of this appeal. What is

19   an output module and how is that being interpreted in the reference? And my

20   understanding of the Hitchcock reference is, as you say, a user can apply to a

21   university and enter in data into a form field, some sort of data entry box,

22   whatever you want to call it, and that gets stored in a database online and then

23   that gets filled in other forms, say, for example.

24   But my understanding of the Examiner's position here is that the

25   Examiner is saying if I make changes to that data in the text box, that the

1 resulting changes are going to be reflected in a new form that gets constructed.

2 And that new form that gets constructed has to be built by software.

3   So the question then becomes, has the "output module" changed in any

4 way? That is to say, has it been invalidated? And I think the Examiner is

5 taking the position that the fact that I'm building a form with new data as

6 opposed to old data means that I'm invalidating the old module, if you will --

7 just for the sake of argument, we'll call it a module for now -- in favor of the

8 new one.

9   And I'm regenerating, if you will, the module with the -- to reflect the

10 new data. And so I think the Examiner is equating the actual software that

11 goes into construction of the form itself.

12   And so my question to you is, is your position based on the fact that

13 the -- it is the data itself that is causing this change to occur and not any

14 software change?

15   MR. HARRIER: Well, I think the difference that I'd like to draw would

16 be, the distinction would be that our -- the output module that we're claiming is

17 actually responsible for the presentation. It is a software module of executable

18 code that another application can then call. And that is responsible for the

19 presentation of the form which then data can be entered into.

20   JUDGE JEFFERY: Right.

21   MR. HARRIER: The Examiner is trying to argue that the actual data, so

22 say a phone number we argued in our Brief. So there would be a phone

23 number entered into the form from Hitchcock, and a user subsequently would

24 like to change that phone number so they do.

1 And then that changes the phone number in the database. So then the

2 next time that an applicant would apply, a new phone number would show up

3 in the form.

4 JUDGE JEFFERY: Wouldn't there be a change though -- the way this

5 system works, though, in Hitchcock is it uses an application data file, does it

6 not?

7 MR. HARRIER: Exactly.

8 JUDGE JEFFERY: I'm talking about the forms engine itself. It takes

9 advantage of what he calls an "application data file," and it is described in

10 detail in paragraph 80 of Hitchcock. And what strikes me about the

11 application data file is that it is a text file, but it has what he calls directives

12 and arguments associated with it that tell the forms engine how to build the

13 form, in effect, in accordance with the data that it has or it can access.

14 And so -- and it actually gives an example in paragraph 80 about a

15 social security number argument or directive, whatever you want to call it, that

16 it interprets. This -- the bottom line here for me is this application data file to

17 me seems like it is an -- it is something that can be interpreted much like code.

18 And it goes to a table to access data, personalized data associated with

19 the particular applicant. But this application data file then tells the system to

20 insert the data into the -- into that particular text box or whatever.

21 So the real mechanics of it is the functionality of the application data file

22 in connection with the database and then that gets parsed to create an HTML

23 form. And the HTML form is then rendered by a browser to provide the look

24 and feel of the form itself to actually build the form on the user's screen.

1   The point of all this is it sounds like to me the Examiner is taking the
2 position that it is not just the data, because the Examiner actually says, as I
3 recall from the Answer, uses the term current "form," to characterize the
4 output module.
5   So the current form is in an HTML format that was built by the forms
6 engine using the application data file in connection with the user attribute
7 tables. So there must be some sort of change in this code, if you will, that -- to
8 build the new look and feel of the new form with the new data. That is all I'm
9 saying.
10   JUDGE COURTENAY: And in your specification, I'm looking at
11 paragraph 6 on page 2 that you have offered as support for the claimed output
12 module, you disclosed the form output module processes the imported
13 application-specific data and its form description data for a presentation via
14 school, printer in parentheses, fax, e-mail, web browser, etc. A web browser
15 uses an HTML form.
16   MR. HARRIER: Sure, sure. Well, our output modules, as I stated
17 before are -- they become reprogrammed. They become invalidated and then
18 will be reprogrammed if and only if it is invalidated. If it is not invalidated,
19 then they can continue to use the form -- or the output module to generate the
20 forms.
21   The closest thing to an output module in Hitchcock would be the form
22 engine that you described, and in Hitchcock in several places including the
23 abstract in paragraph 65 explicitly state that that is never reprogrammed and it
24 is never changed.

1   JUDGE JEFFERY: Why can't HTML that is renderable by a browser to

2 produce a form on my screen, why is that not an output module? I realize --

3 one of the positions you take is it can't be called by a program. But HTML is

4 code, isn't it?

5   MR. HARRIER: HTML, it is a markup language, and here they have

6 this application data file which is a text file, which is read by the forms engine

7 and then the form's engine in turn generates an HTML file that can be read by

8 a browser.

9   JUDGE JEFFERY: So the application data file is -- it is lines of code,

10 in effect, or some sort of text that the forms engine uses that interprets the

11 text to create this HTML. And the HTML in turn provides instructions to

12 the browser to render the form a certain way.

13   MR. HARRIER: Yeah, that is correct. I would agree with you there.

14   JUDGE JEFFERY: Why is that not an output module?

15   MR. HARRIER: The actual HTML?

16   JUDGE JEFFERY: Both. HTML and the application data file.

17   MR. HARRIER: And the application data file? Well, the application

18 data file -- well, another distinction I like to draw between the application data

19 file and our invention is that there is no reusable form elements, which is the

20 entire purpose of our invention. We use -- we have reusable form elements

21 that we might want to use in a hundred different forms.

22   In Hitchcock, the application data files are specific to each, using their

23 example, university that would have an application. And they're not reusing

24 form elements between each application. They make up their own customized

25 data file which is then interpreted by the form.

1   JUDGE JEFFERY: Why can't a form element be a text box where I put

2 my name or social security number in the text box? Isn't that a form element?

3   MR. HARRIER: Well, if you look at our specification at the beginning,

4 our software system wants to go above and beyond simply regenerating text or

5 maybe even a text box, which it can be used by reference. The aim of our

6 invention is to improve that by being able to generate reusable form elements

7 that are more advanced than that which would be, for instance, logic for the

8 entire form, multiple pages or elements within the form that identify the

9 incorporation at the top.

10   So to do that in our system, we can't simply use a reference. We're

11 trying to improve above and beyond the solution in Hitchcock which is why

12 we need an output module which is a specific -- it is a specific module of

13 executable code that is able to then present a form.

14   JUDGE COURTENAY: So your interpretation consistent with your

15 specification is the output module is an executable code, it is a program that

16 processes something?

17   MR. HARRIER: Yes.

18   JUDGE COURTENAY: Does your claim language preclude that?

19 Could not an output module broadly but reasonably read on just a data

20 module?

21   MR. HARRIER: Well, I think that we clearly define it within the

22 specification such that I think a reasonable interpretation would be an

23 executable module as we've argued.

1   JUDGE COURTENAY: So your position is if we interpreted an output
2   module as reading broadly on data as opposed to an executable application that
3   processes data, that would be inconsistent with your specification?
4   MR. HARRIER: I believe so, yes.
5   JUDGE JEFFERY: Well, that begs the question, then, about the
6   application data file itself. Why is the application data file not executable code
7   or at least code that can be interpreted by the forms engine? Because again,
8   when I go to paragraph 80, I see that it tells me the application data file has a
9   series of "directives" and "arguments" that tell the forms engine to do a certain
10   thing. Why is that not code?
11   MR. HARRIER: I think that I would look in the application data file in
12   Hitchcock to our reusable form elements, which are then processed by output
13   module. So in order for Hitchcock to read on our application and our claimed
14   subject matter, I would say that the application data file would be our reusable
15   form elements. And when our reusable form elements change, the output
16   module that is associated with them is invalidated and then is regenerated by a
17   realtime manager whenever it is called the next time.
18   JUDGE JEFFERY: Fair enough. But you would agree that the
19   application data file is executable code or it is at least interpretable code? That
20   is my question to you.
21   MR. HARRIER: No. I would say that application data file, it is a text
22   file that is used by the form engine to generate a form.
23   JUDGE JEFFERY: Well, how is that any different from a BASIC
24   program that lists, like, a number of steps in a program that gets converted into
25   machine language inside the confines of a computer, but itself is just a mere

9

1   text listing of steps that instruct the computer to do something? It seems like

2   there is an analogy here.

3   MR. HARRIER: I think it is different in the sense that an executable

4   module such as our output module is actually like a small block of code that

5   can be called by other applications, and it needs to be rewritten -- that block of

6   code would need to be rewritten if a form element that is commonly used

7   changes.

8   As opposed to an application data file, it is just a text file that is

9   interpreted by and then it is turned into an HTML file, which is then displayed

10   by a browser. So I think that is a distinction I would like to draw.

11   JUDGE JEFFERY: Counsel, you have anything further?

12   MR. HARRIER: Sure. Well, I mean, I would like to address the --

13   specifically the two limitations that the Examiner has rejected. The first is the

14   invalidating of the output modules. I'd like to emphasize the fact that their

15   forms engine, and as I mentioned earlier in paragraph 65 in the abstract, it is

16   never reprogrammed and it is never changed, and nowhere in Hitchcock is

17   something invalidated.

18   So even if the application data file is interpreted to be an output module,

19   at no point in Hitchcock is it invalidated and then regenerated. There is no

20   step of regeneration.

21   JUDGE JEFFERY: Well, what do you make -- if I may, Counsel, the

22   Examiner, though, takes the position that invalidating is sort of an abstract

23   notion, if you will. You know, invalidating -- he actually uses the word

24   "abstract" actually now that I'm looking at it on page 14 of the Answer.

1   The Examiner's position is basically the fact that if I'm going from

2   something old, that it is now "invalid," in effect, and that I'm not going to

3   recreate something based on old, now "invalid," if you will, data. So thereby it

4   invalidated the output module that would have produced the now invalid form,

5   if you will, with the old data. Only the new data is the one that is valid.

6   So that is the position the Examiner seems to be taking here. So unless

7   there is some special meaning of "invalidation" that is imparted to this, why is

8   that incorrect?

9   MR. HARRIER: Well, we disagree with the Examiner and the assertion

10 that there should be no patentable weight given to the term "invalidating" or

11 "invalid" because it is abstract.

12   JUDGE JEFFERY: I don't think he is saying that. I think the

13 Examiner is simply pointing out that it is an abstract term, that it doesn't

14 preclude the fact that the system can abandon one old form construction in

15 favor of a newer form construction.

16   And in that sense, it is "invalidated"—the output module that would

17 have produced the older, now invalid, form. That seems to be the Examiner's

18 position. I don't think he is ignoring the term "invalidate"; he just simply

19 points out that it is abstract.

20   MR. HARRIER: Well, we don't agree with his contention that it is

21 abstract because our specification -- we describe the invalidation and the

22 dependent claims that go towards those. It would actually be a solid step of

23 changing a flag from true to false. And it is not just an abstract idea that we're

24 invalidating. I think the Examiner would like to argue that --

25   JUDGE COURTENAY: But you don't have a flag in claim 1 anyway.

1   MR. HARRIER: On claim 1, the flag is not in there. But the
2   specification defines this step of invalidation.
3   JUDGE COURTENAY: Can you point us to that in your spec? What
4   page is that?
5   MR. HARRIER: Oh, sure.
6   JUDGE COURTENAY: I do see a disclosure in paragraph 29 regarding
7   the testing of the output module. Page 5.
8   MR. HARRIER: Oh, yeah. We do discuss it in paragraph 29. And I
9   was looking for another section. If you look at paragraph 41.
10   JUDGE JEFFERY: Yes, 41.
11   MR. HARRIER: We discuss a valid flag, which may be represented by
12   a Boolean true or false designating whether a reusable form element associated
13   with the output module has been changed since the last activation. An
14   important point is that in Hitchcock when a phone number is changed, for
15   instance, the phone number is just deleted and then replaced with a new one.
16   Our output module is actually to stay intact as a software module in the system
17   with an invalid flag, and not until it is called again would it actually be
18   regenerated and into the proper form.
19   JUDGE JEFFERY: I understand. And that is clearly different in that
20   sense from what is going on in Hitchcock; however, the question then
21   becomes, do we nonetheless still have a difference in the underlying code to
22   cause to generate the new form with the new data as opposed to the old form
23   with the old data? And in that sense, is that meeting "invalidating the module"
24   based on that change?

1   And I think that is -- I understand how the specification describes

2   invalidation in terms of setting and clearing flags and that kind of thing, but I

3   don't see it in the claim.

4   JUDGE COURTENAY: It seems to me that this valid flag in paragraph

5   41 it represents a Boolean true-false value that designates whether a reusable

6   form element associated with the output module has been changed. And your

7   claim recites invalidating the module itself. Your specification discloses the

8   flag is associated with this reusable form element.

9   MR. HARRIER: Well, the flags are --

10   JUDGE COURTENAY: There seems to be a distinction.

11   MR. HARRIER: They're associated with both the reasonable form

12   elements and then a reasonable form element that is then associated with an

13   output module so that when a reasonable form element is changed, all output

14   modules that utilize that reusable form element would then be invalid.

15   JUDGE COURTENAY: So it is an indirect association?

16   JUDGE JEFFERY: Yeah. Change occurs. Flags get set or cleared,

17   whatever.

18   MR. HARRIER: Within all output modules.

19   JUDGE JEFFERY: There is a cause and effect, in effect?

20   MR. HARRIER: Yeah.

21   JUDGE JEFFERY: Okay. Anything else? Thank you, counsel.

22   MR. HARRIER: Thank you very much for your time.

23

24   (Whereupon, the proceedings were concluded on Wednesday,

25   September 9, 2009.)